APPLICATION

REPORT

Hybrid Computer Solution Of Two Point Boundary Value Problems

By Edward J. Fadden

APPLIED DYNAMICS

Introduction

The analog computer has proved to be a very valuable tool in solving engineering problems. Many of these computational problems require the use of trial-and-error or searching procedures if solutions are to be obtained. The use of conventional trial-and-error techniques by the computer operator is often quite time consuming, particularly if a large number of solutions is required.

In recent years "hybrid" computers have been developed which employ both analog and digital elements. The searching operation required in solving problems of the type mentioned above can often be performed automatically and at high speed on a hybrid computer. If the searching operation is to be implemented entirely by the computer, it is necessary to specify a sequence of operations which the computer is programmed to perform repeatedly. The terms "iterative procedure" and "algorithm" are used to denote this sequence of operations.

Two-point boundary value problems belong to the class of problems for which searching procedures are required. In this paper a simple example of a two-point boundary value problem is considered. An iterative procedure is devised for this example and a computer circuit for implementing this procedure is described. Numerical results obtained by using this circuit are presented. This example provides a convenient means for illustrating the mathematical difficulties which arise in the use of iterative procedures.

The Example Problem

For the one-dimensional flow of heat in a slab, the temperature distribution is determined by solving the equation,

$$C(x)\,\frac{\partial\theta(x,\tau)}{\partial\tau} = \frac{\partial}{\partial x}\left[K(x)\,\frac{\partial\theta(x,\tau)}{\partial x}\right] + f(x,\tau), \qquad (1)$$

subject to appropriate initial and boundary conditions, where:

$\theta(x, \tau)$ = temperature,

$C(x)$ = specific heat,

$K(x)$ = thermal conductivity,

x = distance along the slab, $0 \le x \le 1$,

τ = time, and

$f(x, \tau)$ = rate of heat flow per unit volume into or out of the medium.

Equation (1) is a linear partial differential equation. For the purposes of this paper it will be assumed that $f(x, \tau) \equiv 0$. The method of separation of variables will now be used to separate equation (1) into two ordinary, linear differential equations. In this method a solution of the form

$$\theta(x, \tau) = X(x)T(\tau) \tag{2}$$

is assumed. Substituting (2) into (1) and dividing the resulting equation by $X(x)T(\tau)$ gives

$$[T(\tau)]^{-1} \frac{dT(\tau)}{d\tau} = [C(x)X(x)]^{-1} \frac{d}{dx}\left[K(x) \frac{dX(x)}{dx}\right] . \tag{3}$$

The only way in which this equation can be satisfied as x and τ vary is for each side of this equation to be equal to a constant, say α. It can easily be shown from physical considerations that α must be negative. Setting $\lambda = -\alpha$ (so that $\lambda > 0$), the following equations are obtained from (3):

$$\frac{dT}{d\tau} + \lambda T = 0, \tag{4}$$

$$\frac{d}{dx}\left[K(x) \frac{dX}{dx}\right] + \lambda C(x)X = 0. \tag{5}$$

The solution of equation (4) is straightforward. However, equation (5) must be solved subject to the boundary conditions at

$x = 0$ and $x = 1$. For simplicity, assume that the boundary conditions are

$$X(0) = X(1) = 0. \tag{6}$$

The solutions of equation (5) will only satisfy the prescribed boundary conditions (6) for certain discrete values of λ, say λ_n, where $n = 1, 2, \cdots$. The numbers λ_n are called eigenvalues. Let $X_n(x)$ denote a solution of equation (5) which is obtained using $\lambda = \lambda_n$. Then $X_n(x)$ satisfies the boundary conditions (6). The function $X_n(x)$ is termed an eigenfunction or normal mode.

These concepts will now be illustrated by considering the case where the slab is homogeneous, i.e., where $C(x)$ and $K(x)$ are constant for $0 \leq x \leq 1$. Suppose that $C(x) = K(x) \equiv 1$. For this case equation (5) becomes

$$\frac{d^2X}{dx^2} + \lambda X = 0 \tag{7}$$

and the solution of this equation is

$$X(x) = A \sin \sqrt{\lambda}x + B \cos \sqrt{\lambda}x.$$

Since $X(0) = 0$, it follows that $B = 0$. The other boundary condition requires that $A \sin \sqrt{\lambda} = 0$. In order to obtain a non-trivial solution, it is necessary that $\sin \sqrt{\lambda} = 0$, or $\sqrt{\lambda} = n\pi$. Hence the eigenvalues for this case are $\lambda_n = (n\pi)^2$, $n = 1, 2, \cdots$. The function $X_m(x) = A \sin \sqrt{\lambda_m}\, x$ is the mth normal mode with amplitude A.

The problem to be solved is the following: Find the first N eigenvalues and corresponding normal modes of equation (5) subject to the boundary conditions given by (6).

The Proposed Method of Solution

Once the problem has been stated in mathematical terms, the next step is the formulation of a logical procedure to be programmed

on the computer. A procedure will now be devised for the problem described above.

Consider the solution of equation (5) on the hybrid computer. The independent variable in the computer solution is computer time. The boundary condition $X(0) = 0$ becomes an initial condition for the computer solution. The other boundary condition, $X(1) = 0$, must be satisfied by trial-and-error adjustment of the parameter λ. Note that the choice of $\left.\frac{dX}{dx}\right|_{x=0}$, which is needed for the computer solution, is arbitrary, since the value of $\left.\frac{dX}{dx}\right|_{x=0}$ affects only the amplitude and not the shape of the solution $X(x)$.

Suppose that the computer is seeking the eigenvalue λ_n. The computer must use some criterion in correcting the parameter λ on each iteration to insure that when $|X(1)| < \epsilon$ (ϵ small determines the accuracy to which the boundary condition is satisfied), $X(x)$ is close to $X_n(x)$ and not some other normal mode. The criterion to be used is based on the zero-crossing property of the normal modes: namely, the larger the eigenvalue λ_n, the closer together lie the zeros of the normal mode, $X_n(x)$. This property will now be shown. *

Let λ_j and λ_k be two distinct eigenvalues of equation (5) for the boundary conditions (6); let $X_j(x)$ be a solution of equation (5) with $\lambda = \lambda_j$; and let $X_k(x)$ be a solution of equation (5) with $\lambda = \lambda_k$. (Note: This proof is not restricted to the boundary conditions given by (6), but holds for most of the usual boundary conditions.) Multiplying the equation for $X_j(x)$ by $X_k(x)$ and that of $X_k(x)$ by $X_j(x)$ and subtracting the second of these equations from the first yields

$$X_k\left[K(x)\frac{dX_j}{dx}\right] + \lambda_j C(x) X_j X_k - X_j\left[K(x)\frac{dX_k}{dx}\right] - \lambda_k C(x) X_j X_k = 0,$$

*For a more detailed discussion of this and related properties, see reference [1].

or

$$\frac{d}{dx}\left[K(x)\left(X_k \frac{dX_j}{dx} - X_j \frac{dX_k}{dx}\right)\right] = (\lambda_k - \lambda_j)\, C(x)\, X_j X_k . \qquad (8)$$

For some range of values of λ the solution, $X(x)$, of equation (5) goes through zero at least once for $0 < x < 1$. Let λ_j lie in this range, and let a be the smallest value of $x > 0$ such that $X_j(x) = 0$. Clearly $a < 1$. Integrating equation (8) from $x = 0$ to $x = a$ and applying the conditions $X_j(0) = X_k(0) = X_j(a) = 0$ gives the equation

$$X_k\, K(x) \left.\frac{dX_j}{dx}\right|_{x=a} = (\lambda_k - \lambda_j) \int_0^a C(\sigma) X_j(\sigma) X_k(\sigma)\, d\sigma . \qquad (9)$$

Suppose that $X_j(x) > 0$ for $0 \le x \le a$. Then $\left.\frac{dX_j}{dx}\right|_{x=a} < 0$. Also suppose that $X_k(x)$ does not change sign for $0 \le x \le a$. For the sake of argument let $X_k(x)$ be positive for $0 \le x \le a$. Now $K(x) > 0$ and $C(x) > 0$ for $0 \le x \le 1$. Inspection of equation (9) shows that, under the above hypothesis, the right hand side is positive while the left hand side is negative. This is impossible. Hence $X_k(x)$ must change sign at least once for $0 \le x \le a$. This same argument can be used to show that $X_k(x)$ must change sign between any two successive zero-crossings of $X_j(x)$.

Let the eigenvalues be ordered numerically so that $0 < \lambda_1 < \lambda_2 < \lambda_3 < \cdots$. (This ordering is possible, since the eigenvalues are all real.) Then, if $X_n(x)$ and $X_{n+1}(x)$ are solutions of equation (5) with $\lambda = \lambda_n$ and $\lambda = \lambda_{n+1}$ respectively, it follows from the above argument that $X_{n+1}(x)$ has at least one more zero-crossing than $X_n(x)$ for $0 \le x \le 1$. Since the solution, $X(x)$, of equation (5) depends continuously on λ, it follows that $X_{n+1}(x)$ can have at most one more zero-crossing than $X_n(x)$. If this were not the case, then there would be a value of λ, say $\bar{\lambda}$, such that $\lambda_n < \bar{\lambda} < \lambda_{n+1}$ and the solution of equation (5) with $\lambda = \bar{\lambda}$ would satisfy the boundary conditions (6), i. e., $\bar{\lambda}$ would be an

eigenvalue. But this contradicts the fact that λ_n and λ_{n+1} are successive eivenvalues. Hence $X_{n+1}(x)$ has just one more zero-crossing than $X_n(x)$ for $0 \leq x \leq 1$. It follows from the definition of an eigenvalue and the fact that the eigenvalues are ordered, that $X_n(x)$ has n zero-crossings for $0 < x \leq 1$. (Note that $X_n(0) = 0$ is not considered as a zero-crossing of $X_n(x)$.)

Suppose that the nth eigenvalue, λ_n, and a corresponding normal mode, $X_n(x)$, are to be determined. Let λ_n^i denote the ith approximation to λ_n; let $X_n^i(x)$ be a solution of equation (5) obtained with $\lambda = \lambda_n^i$; and let x_n^i denote the value of x for which the nth zero-crossing of $X_n^i(x)$ occurs (i.e., let x_n^i be the nth positive root of the equation $X_n^i(x) = 0$). Note that if $\lambda_n^i = \lambda_n$, then $x_n^i = 1$. It follows from the discussion of the zero-crossing property of the normal modes that, if $x_n^i > 1$, then $\lambda_n^i < \lambda_n$ and hence λ_n^{i+1} (i.e., the (i + 1) approximation to λ_n) should be chosen such that $\lambda_n^{i+1} > \lambda_n^i$. Likewise, if $x_n^i < 1$, then $\lambda_n^i > \lambda_n$ and λ_n^{i+1} should be chosen so that $\lambda_n^{i+1} < \lambda_n^i$. This suggests that one way of choosing λ_n^{i+1} is by means of the recurrence relation

$$\lambda_n^{i+1} = \lambda_n^i + k(x_n^i - 1) , \qquad (10)$$

where $k > 0$ is a constant.

The problem proposed earlier is that of determining the first N eigenvalues and corresponding normal modes. The iterative procedure to be considered is based on equation (10). In this procedure the computer must perform the following sequence of operations:

1) Store the number n in a reference counter;
2) If $n > N$, stop the program; otherwise proceed to step 3);
3) Solve equation (5) with $\lambda = \lambda_n^i$ for $X_n^i(x)$ subject to the initial conditions $X_n^i(0) = 0$ and $\left.\frac{dX_n^i}{dx}\right|_{x=0} = A$ where $A \neq 0$;
4) Count the zero-crossings of $X_n^i(x)$ and compare this number

to the number n stored in the reference counter to determine x_n^i;

5) If $|x_n^i - 1| > \epsilon$ (where $\epsilon > 0$ is specified a priori), use equation (10) to obtain λ_n^{i+1} and repeat steps 3), 4), and 5.);

6) If $|x_n^i - 1| \leq \epsilon$, record λ_n^i and $X_n^i(x)$ for $0 \leq x \leq 1$, add one to the number stored in the reference counter, and return to step 2).

Convergence of the Iterative Procedure

The iterative procedure for determining λ_n generates a sequence $(\lambda_n^1, \lambda_n^2, \ldots, \lambda_n^i)$ of approximations to λ_n. An important question to be considered is whether or not this sequence actually converges to λ_n. Useful information can be gained by linearizing equation (10) about λ_n.

Let $X(x)$ be a solution of equation (5) for some positive value of the parameter λ. Suppose that the equation $X(x) = 0$ has at least m positive roots for $0 < x < \infty$ where $m \geq 1$. Let x_j denote the jth positive root of $X(x) = 0$ where $1 \leq j \leq m$. It was pointed out earlier that $X(x)$ depends continuously on λ. It follows from this fact that $x_j = x_j(\lambda)$ is a continuous function of λ. This is true for each j. Hence $x_n^i = x_n(\lambda_n^i)$ and $x_n(\lambda_n) = 1$.

Suppose that $x_n(\lambda)$ has a Taylor's series expansion about the point $\lambda = \lambda_n$. Then

$$x_n(\lambda) = 1 + \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n} (\lambda - \lambda_n) + r(\lambda - \lambda_n), \tag{11}$$

where $(\lambda - \lambda_n)^{-1} r(\lambda - \lambda_n) \to 0$ as $\lambda \to \lambda_n$. Substituting equation (11) with $\lambda = \lambda_n^i$ into equation (10) gives

$$\lambda_n^{i+1} = \lambda_n^i + k \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n} (\lambda_n^i - \lambda_n) + kr(\lambda_n^i - \lambda_n), \tag{12}$$

so that the linearized difference equation corresponding to equation (10) is

$$\delta\lambda_n^{i+1} = \left[1 + k \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n}\right] \delta\lambda_n^i , \tag{13}$$

where $\delta\lambda_n^i = \lambda_n^i - \lambda_n$. The solution of equation (13) is

$$\delta\lambda_n^i = \left[1 + k \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n}\right]^i \delta\lambda_n^0 , \tag{14}$$

where λ_n^0 is the initial approximation to λ_n.

It can be shown that, if $\delta\lambda_n^0$ is sufficiently small, then the solution of equation (10) (which has λ_n^0 as its initial value) converges to zero provided that all solutions of equation (13) converge to zero. It is clear from equation (14) that solutions of equation (13) converge to zero if and only if $\left|1 + k \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n}\right| < 1$. Since $\left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n} < 0$, this condition requires that

$$0 < k < 2 \left| \left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n} \right|^{-1} . \tag{15}$$

Thus, if the value of k chosen is too large, the iterative procedure will not converge. On the other hand, if k is chosen conservatively small, the convergence may be quite slow. Furthermore, since $\left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n}$ depends on n, the range of k for which solutions of equation (13) converge may vary significantly for different n.

To clarify these points, the case of the homogeneous slab described by equation (7) will be considered briefly. It follows from the earlier discussion of this case that x_n is determined from the equation

$\sqrt{\lambda}\, x_n = n\pi$. Therefore, $x_n(\lambda) = n\pi(\lambda)^{-\frac{1}{2}}$ and $\left.\frac{dx_n}{d\lambda}\right|_{\lambda=\lambda_n} = -\frac{1}{2}n\pi(\lambda_n)^{-\frac{3}{2}} = -\frac{1}{2}(n\pi)^{-2}$, since $\lambda_n = (n\pi)^2$. For this case, condition (15) takes the form $0 < k < 4(n\pi)^2$. Suppose that $k = 2\pi^2$. Then for $n = 1$, equation (14) shows that $\delta\lambda_1^1 = 0$ for any $\delta\lambda_1^0$. On the other hand, suppose $k = 2\pi^2$ and $n > 1$. Then equation (14) becomes

$$\delta\lambda_n^i = (1 - \frac{1}{n^2})^i \delta\lambda_n^0 ,$$

and it is evident that the convergence is quite slow for large n.

This problem of selecting a useful value of k is not quite as serious from a practical standpoint as the above discussion would seem to indicate. The reason is that the computer can usually be programmed to automatically increase or decrease the value of k (at least within some range of values). Alternatively, the computer operator can make appropriate changes in k whenever such changes are required.

Another question which presents itself is whether or not the boundary condition at $x = 1$ is satisfied with sufficient accuracy when $|x_n^i - 1| \leq \epsilon$ for a specified ϵ. It may well be that $X_n^i(1)$ is relatively large even though $|x_n^i - 1|$ is quite small. A better iterative procedure for this problem would be one based on reducing $|X_n^i(1)|$ directly. However, the computer circuit required for this type of algorithm is more complex than the circuit required for the iterative procedure described earlier.

The Hybrid Computer

Now that an iterative procedure has been formulated for the problem, the next step is to obtain a computer circuit for implementing this procedure. First, however, the hybrid computer will be discussed briefly and the symbols required for programming the iterative procedure will be described.

The elements in a hybrid computer can be categorized as analog, digital, and analog-digital, depending on the class of signals with which the elements deal. For the problem considered in this paper only the patchable digital elements to be described below are required. However, in problems which require numerical calculations and complex schemes for control of the analog-digital elements, a stored program digital computer with flexible input-output routines is needed.

The symbols used for programming the iterative procedure are shown in figure 1. Light-weight lines and lower-case letters are used to designate the analog signals, while heavy-weight lines and upper-case letters are used for logic signals. A logic signal, A, is either in the state A = 1 or in the state A = 0. $\overline{A}$ is used to denote the complement of A, i. e., if A = 1, then $\overline{A}$ = 0 and if A = 0, $\overline{A}$ = 1.

The operation of the elements depicted in figure 1 will now be described.

a) Summer: $+e_0 = \alpha(10e_1 + e_2 + i)$. The standard feedback is 0.1MΩ (i. e., $\alpha = 0.1$). Note that both plus and minus output signals are available. This type of element is called bi-polar.

b) Potentiometer: $e_0 = ke_1$.

c) Multiplier: $i = 10^{-1}xy$.

d) Integrator: Each integrator is separately controlled by the logic signals A, B, C, and D. Electronic switches are used for high speed mode changing.

Reset Mode: A = 0, B = 0, 1; $e_0 = e$.

Operate Mode: A = 1, B = 0; $e_0 = \alpha \int_0^t (10e_1 + e_2 + i)dt + e$.

Hold Mode: A = 1, B = 1.

Time Scale: C = 0, D = 0; real time (integrator gain = α).

C = 1, D = 0; speed up computer time by factor

of 10

C = 0, D = 1; speed up computer time by factor of 100.

The standard feedback is a 1μfd. capacitor (i.e., $\alpha = 1$).

e) Switch (Single pole, double throw):

A = 0; $i = 10e_1$

A = 1; $i = 10e_2$

f) Comparator:

B = 0: A = 1 if $e_1 + e_2 > \epsilon$;

A = 0 if $e_1 + e_2 < -\epsilon$;

A = 0, 1 if $|e_1 + e_2| < \epsilon$, i.e., for $|e_1 + e_2| < \epsilon$, the output A depends on the past history of $e_1 + e_2$ so that A has the value it last had when $|e_1 + e_2| > \epsilon$. (The small dead space ϵ and associated hysteresis loop provides noise suppression).

B = 1: Locks A at value present when B = 0 → 1 where 0 → 1 denotes a transition from logic 0 to logic 1.

g) Track-Transfer (Memory element):

A = 0, B = 0, 1: $e_0 = e$ (Initial condition mode)

A = 1, B = 0, 1: hold e_0 constant

A = 1, B = 0 → 1: transfer $(e_1 + e_2)$ to e_0 and hold e_0 constant until next B = 0 → 1.

h) Input Button:

B = 0: A = 1 if button down,
A = 0 if button up.

B = 1: A = 0, 1, push button to reverse A.

i) Pulser (Variable period):

A = 1, 0: B = 0

A = 0 → 1: B = 1 for T seconds

C = 0, D = 0: T = 1 second

C = 1, D = 0: T = 10 milliseconds

C = 0, D = 1: T = 100 milliseconds

j) Pulser (Fixed period):

k) Flip-flop:

A = 0, B = 0: D = 0, 1;

A = 1, B = 0: D = 1

A = 0, B = 1: D = 0

A = 1, B = 1 not used

A = 0, B = 0, C = 0 → 1: reverse D.

l) Or Gate:

D = 0 only if A = 0, B = 0, and C = 0.

The Computer Program

A block diagram of the proposed computer program is shown in figure 2. The circuits for the various blocks are contained in figures 3 - 8. These circuits are described below. Timing diagrams are included with most of these circuits. These timing diagrams are not only helpful in explaining the behavior of a circuit, but are also useful in designing a computer circuit to perform a specified operation.

In the computer solution of this problem, the independent variable is computer time t. The relationship between t and x is $t = \bar{t}x$ so that $t = \bar{t}$ corresponds to x = 1. Equation (5) can be written in terms of t as

$$\frac{d}{dt}\left[K(\bar{t}^{-1}t)\frac{dy(t)}{dt}\right] + \bar{t}^{-2}\lambda C(\bar{t}^{-1}t)y(t) = 0 \qquad (16)$$

where $y(t) = X(\bar{t}^{-1}t)$.

The on-off control for the program is pushbutton 1. When this

control is in the "off" state (i.e., $\overline{S} = 1$) the integrators are in reset, the track-transfers are in the initial condition mode and the flip-flops are cleared.

The timing circuit shown in figure 3 is the clock which controls the reset and operate periods. It also furnishes the independent variable t for use elsewhere in the circuit. When $\overline{B} = 1$, the integrator goes into the operate mode and generates the ramp ct. For ct < 100 volts, the outputs of C1 and P1 are each logic zero. When ct exceeds 100 volts, the output of C1 changes from 0 to 1 and triggers P1. This in turn puts the integrator in the reset mode. This clock can be interrupted by placing the integrator in the hold mode. This is done by signal D whenever a zero-crossing of $y_n^i(t)$ occurs (where $y_n^i(t)$ is a solution of equation (16) obtained for $\lambda = \lambda_n^i$). The signal F comes from the manual and programmed stop circuit and F = 0 unless a stop has been requested.

The relationship between the various signals is shown in the timing diagram of figure 3.

Figure 4 shows the zero-crossing counter. The reference counter is TT2. The output of TT2 is $10(n - \frac{1}{2})$ volts where n is the number of zero-crossings of $y_n(t)$ for $0 < t \leq \overline{t}$. The initial condition of -5 volts on TT2 is used to insure an accurate switching of C3 when the nth zero-crossing of $y_n^i(t)$ occurs. Once λ_n and $y_n(t)$ have been determined and recorded, the print out routine causes G = 1 for a short period of time. The signal G = 0 → 1 triggers TT2 and its output changes to $10(n + \frac{1}{2})$ volts.

The number of zero-crossings of $y_n^i(t)$ is counted by TT1. On each zero-crossing, the output of C2 changes state and triggers either P2 or P3. Thus for each zero-crossing, D = 1 for a period of several milliseconds. The signal D = 0 → 1 causes 10 volts to be added to the output of TT1. The signal D = 1 is also used to put the clock (figure 3) and the integrators used in the simulation of equation (16) in the hold

mode while TT1 is being updated. This insures an accurate determination of $(t_n^i - \bar{t})$, if $ct_n^i < 100$ volts (where $t_n^i = \bar{t}\, x_n^i$). The output of TT1 is reset to zero during each reset period.

At time $t = t_n^i$ (if $ct_n^i < 100$ volts) the output of C3 changes state, i.e., $A = 0 \rightarrow 1$. This signal is used to initiate the parameter correction routine.

The parameter correction circuit is shown in figure 5. When $|t_n^i - \bar{t}| < \epsilon$, the signal H is logic one and H is used to prevent any change in λ_n^i. However, if $|t_n^i - \bar{t}| \geq \epsilon$ as is generally the case, $H = 0$ and the value of λ is updated when $A = 0 \rightarrow 1$ or $C = 0 \rightarrow 1$. If $ct_n^i < 100$ volts, $A = 0 \rightarrow 1$ when $t = t_n^i$ and this signal causes λ to be updated according to the equation

$$\lambda_n^{i+1} = \lambda_n^i + kc(t_n^i - \bar{t}) . \tag{17}$$

On the other hand, if $ct_n^i \geq 100$ volts, $A = 0$ for $0 \leq ct \leq 100$ volts and $C = 0 \rightarrow 1$ causes λ to be updated as follows:

$$\lambda_n^{i+1} = \lambda_n^i + k(100 - c\bar{t}) . \tag{18}$$

The signal E comes from the automatic scale changing circuit. When $E = 1$ the gain constant k in equations (17) and (18) is replaced by $0.1k$.

Since the problem stated that the first N normal modes are to be determined, the computer must be programmed to stop after Nth mode has been recorded. It is also convenient to have a manual stop button. This, when depressed, causes the computer to stop after it has determined and recorded the normal mode that it is currently seeking. This manual stop is useful during check-out for example. The circuit for performing the stop operation is shown in figure 6.

The end point error test circuit is shown in figure 7. The comparator C5 compares $\frac{1}{2}|t - \bar{t}|$ to ϵ. If $ct_n^i < 100$ volts, $A = 0 \rightarrow 1$

at $t = t_n^i$ and the signal A = 1 is used to latch C5. If $\frac{1}{2}|t_n^i - t| < \epsilon$, the flip-flop is set, i. e., H = 0 → 1, and the print out routine is initiated. For simplicity the details of this operation have been omitted. At the termination of the print out procedure G = 1 for a short period. This signal is used to set H = 0 and update the reference counter, TT2. The signal C = 1 is used to latch C5 during the reset period to insure that false triggering of the flip-flop does not occur.

Figure 8 shows the automatic scale changing circuit. The addition of this circuit allows a significant increase in the number of eigenvalues that can be determined before an overload occurs.

In figure 8 the integrator is used as a track-hold device. It is normally in the track mode. When the output of TT3 exceeds 100 volts, J = 0 → 1. This triggers the pulser which latches C6. The signal J = 1 also sets the flip-flop, puts the integrator in the hold mode, and puts TT3 in the initial condition mode. The output of TT3 is now set to one-tenth of the value it formerly held. The signal E = 1 is used to decrease by a factor of ten the size of the integrating capacitors in the circuit which solves equation (16) and to decrease the gain in the parameter correction circuit by a factor of ten. The signals A and C are used to latch C6 so that the scale changing does not occur while TT3 is in the transfer mode. Additional circuitry is required if it is desired to make another scale change when and if the output of TT3 exceeds 100 volts for the second time.

Numerical Results

Numerical results were obtained for the case of the homogeneous slab described by equation (7). For this problem equation (16) reduces to

$$\frac{d^2y}{dt^2} + \bar{t}^{-2}\lambda y = 0. \qquad (19)$$

The circuit for solving this equation is shown in figure 9. To obtain

good amplitude scaling, $100\bar{t}^{-1}\sqrt{\lambda_n^i}$ is used as the parameter input signal to each of the multipliers. Let $\gamma_n^i = \bar{t}^{-1}\sqrt{\lambda_n^i}$. For this problem it is convenient to modify the iterative procedure slightly. The computer was programmed to seek γ_n rather than $\bar{t}^{-2}\lambda_n$. The iterative procedure used was based on the equation

$$\gamma_n^{i+1} = \gamma_n^i + kc(t_n^i - \bar{t}). \tag{20}$$

It is helpful at this point to linearize equation (20) about γ_n. Since the derivation of this linearized equation parallels the derivation of equation (13), the details are omitted. The desired linearized equation is

$$\delta\gamma_n^{i+1} = \left[1 + kc\left.\frac{dt_n}{d\gamma}\right|_{\gamma=\gamma_n}\right]\delta\gamma_n^i , \tag{21}$$

where $\delta\gamma_n^i = \gamma_n^i - \gamma_n$. All solutions of this equation converge to zero if $0 < k < 2\left|c\left.\frac{dt_n}{d\gamma}\right|_{\gamma=\gamma_n}\right|^{-1}$. Since $t_n(\gamma) = n\pi(\gamma)^{-1}$, $\left.\frac{dt_n}{d\gamma}\right|_{\gamma=\gamma_n} = -n\pi(\gamma_n)^{-2}$ $= -\bar{t}^2(n\pi)^{-1}$ (using the fact that $\gamma_n = n\pi(\bar{t})^{-1}$). Thus the above condition on k can be written as $0 < k < 2\left|-c\bar{t}^2(n\pi)^{-1}\right|^{-1}$. In obtaining the numerical results the values c = 6.667 volts/sec and $\bar{t}$ = 10 sec. were used. For these values of c and $\bar{t}$ and for n = 1, it is necessary that $0 < k < 2n\pi(c\bar{t}^2)^{-1}$ = 0.00942 if all the solutions of equation (21) are to converge to zero. The value of k actually used was k = 0.006.

Substituting the above values for c, k, and $\left.\frac{dt_n}{d\gamma}\right|_{\gamma=\gamma_n}$ into equation (21) yields

$$\delta\gamma_n^{i+1} = [1 - 1.274(n)^{-1}]\delta\gamma_n^i , \tag{22}$$

and the solution of this equation for the initial value $\delta\gamma_n^0$ is

$$\delta\gamma_n^i = [1 - 1.274(n)^{-1}]^i \delta\gamma_n^0 .$$

This shows that the number of iterations required to obtain a solution of specified accuracy will increase as n increases.

The numerical results obtained for this problem are presented in Table 1. Inspection of the data shows that equation (22) gives a good qualitative indication of the type of behavior to be expected for γ_n^i in the vicinity of γ_n.

Conclusions

In engineering applications computational problems often arise which require the use of trial-and-error procedures if solutions are to be obtained. For many problems of this type conventional trial-and-error methods can be replaced by iterative routines which can be implemented on a hybrid computer system.

In this paper a simple example is considered. An iterative procedure is formulated and the mathematical difficulties which arise in the use of this procedure are discussed. A computer circuit for implementing this iterative routine is described and some numerical results obtained by using this circuit are presented.

Reference 1

P.M. Morse and H. Feshback, "Methods of Theoretical Physics," Part I, pp 719-724, McGraw-Hill Co., Inc., 1953.

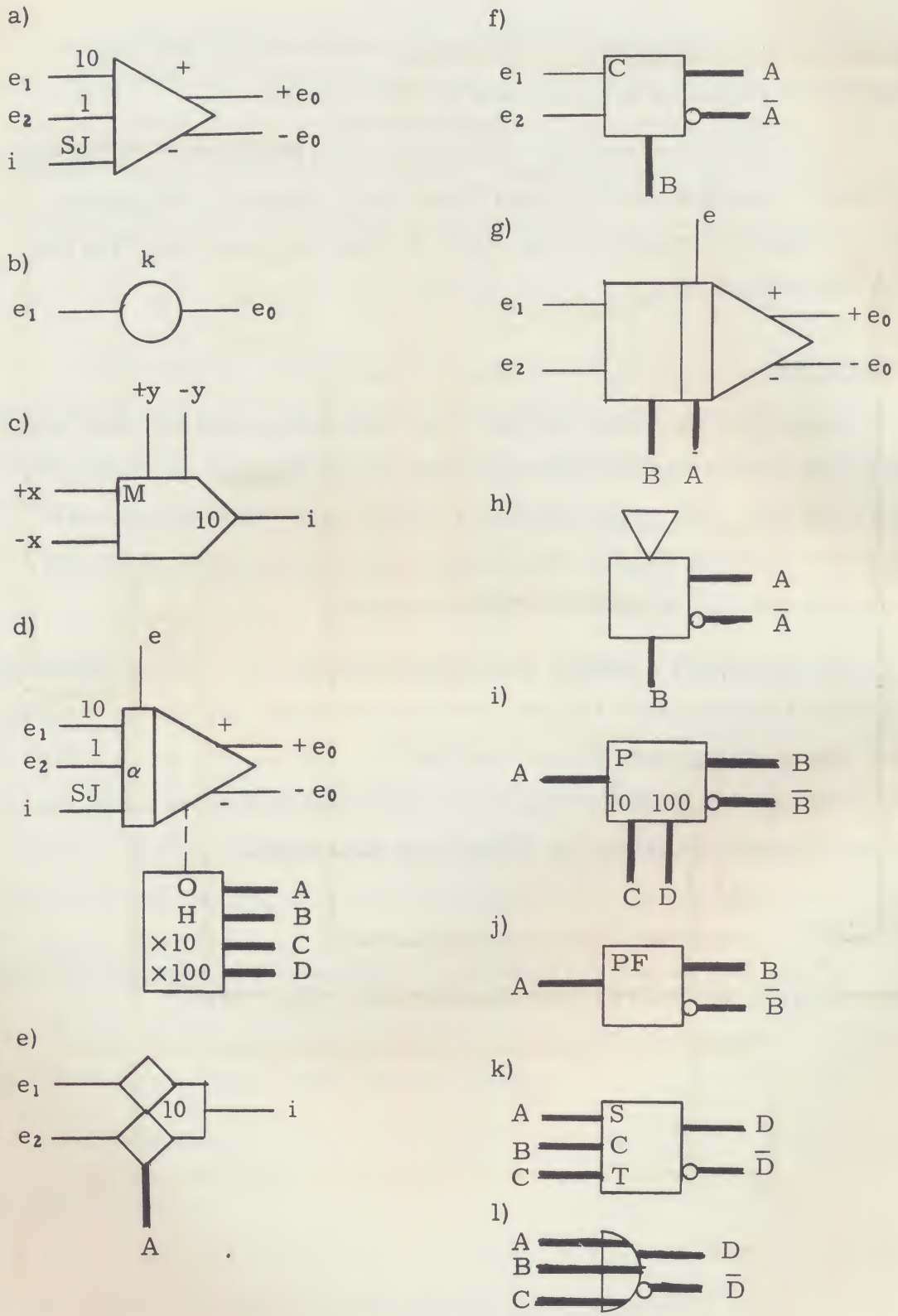
a)
10
1
SJ
e_1
e_2
i
+
-
$+e_0$
$-e_0$
b)
k
e_1
e_0
c)
+y -y
+x
-x
M
10
i
d)
e
10
1
SJ
e_1
e_2
i
α
+
-
$+e_0$
$-e_0$
O
H
×10
×100
A
B
C
D
e)
e_1
e_2
10
i
A
f)
e_1
e_2
C
A
$\overline{A}$
B
g)
e
e_1
e_2
+
-
$+e_0$
$-e_0$
B A
h)
A
$\overline{A}$
B
i)
A
P
10 100
B
$\overline{B}$
C D
j)
A
PF
B
$\overline{B}$
k)
A
B
C
S
C
T
D
$\overline{D}$
l)
A
B
C
D
$\overline{D}$

Figure 1

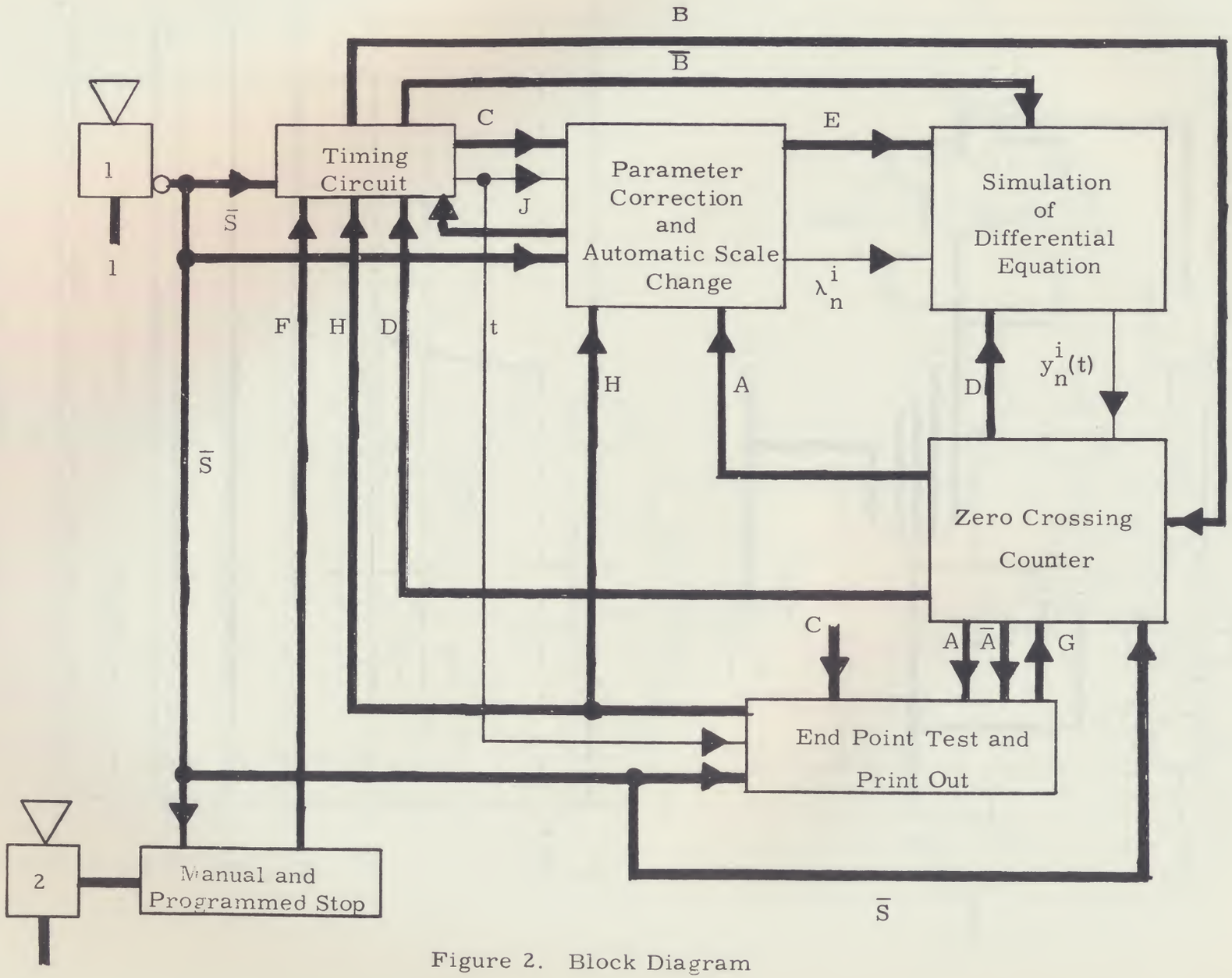

Figure 2. Block Diagram

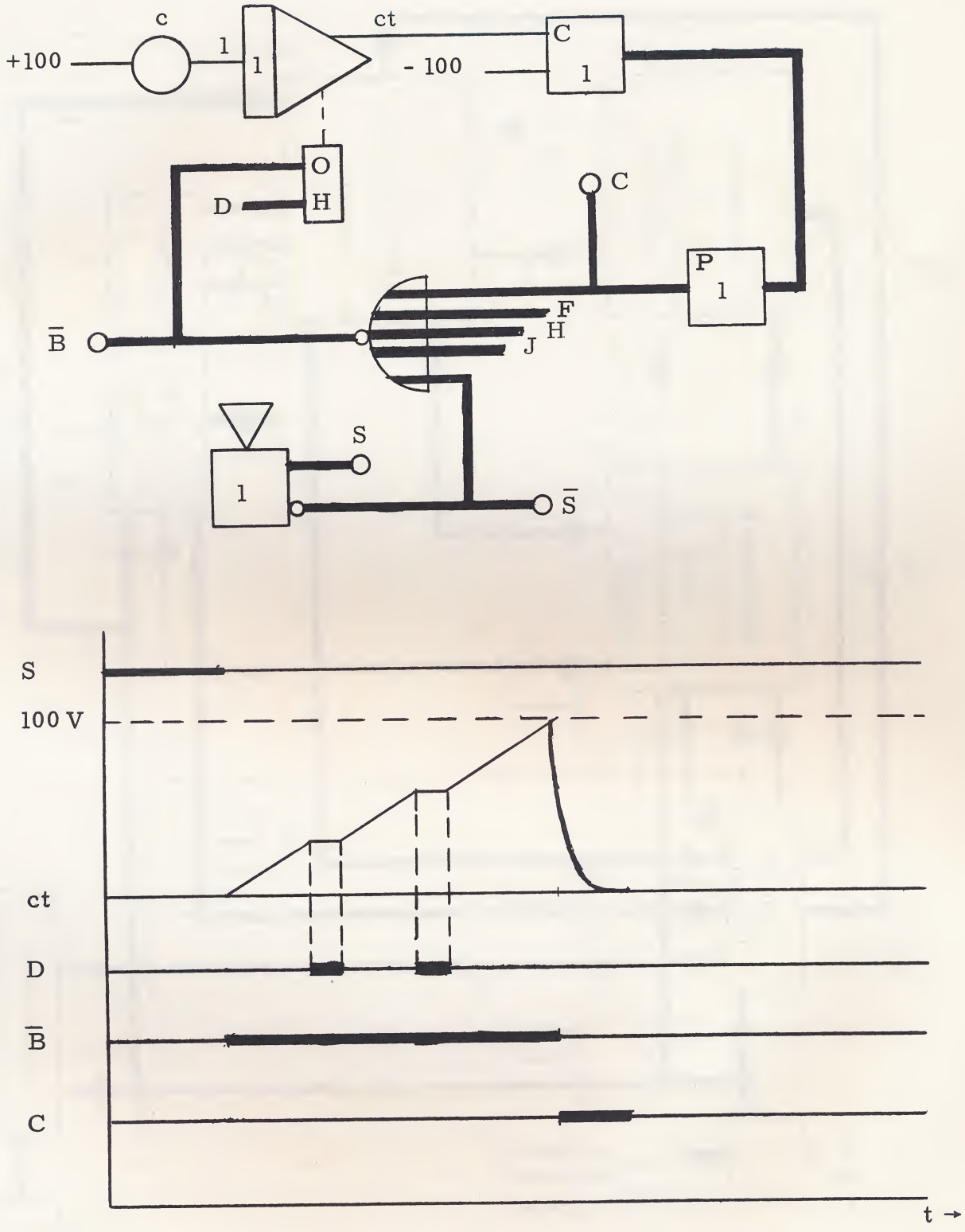

Figure 3. Timing Circuit

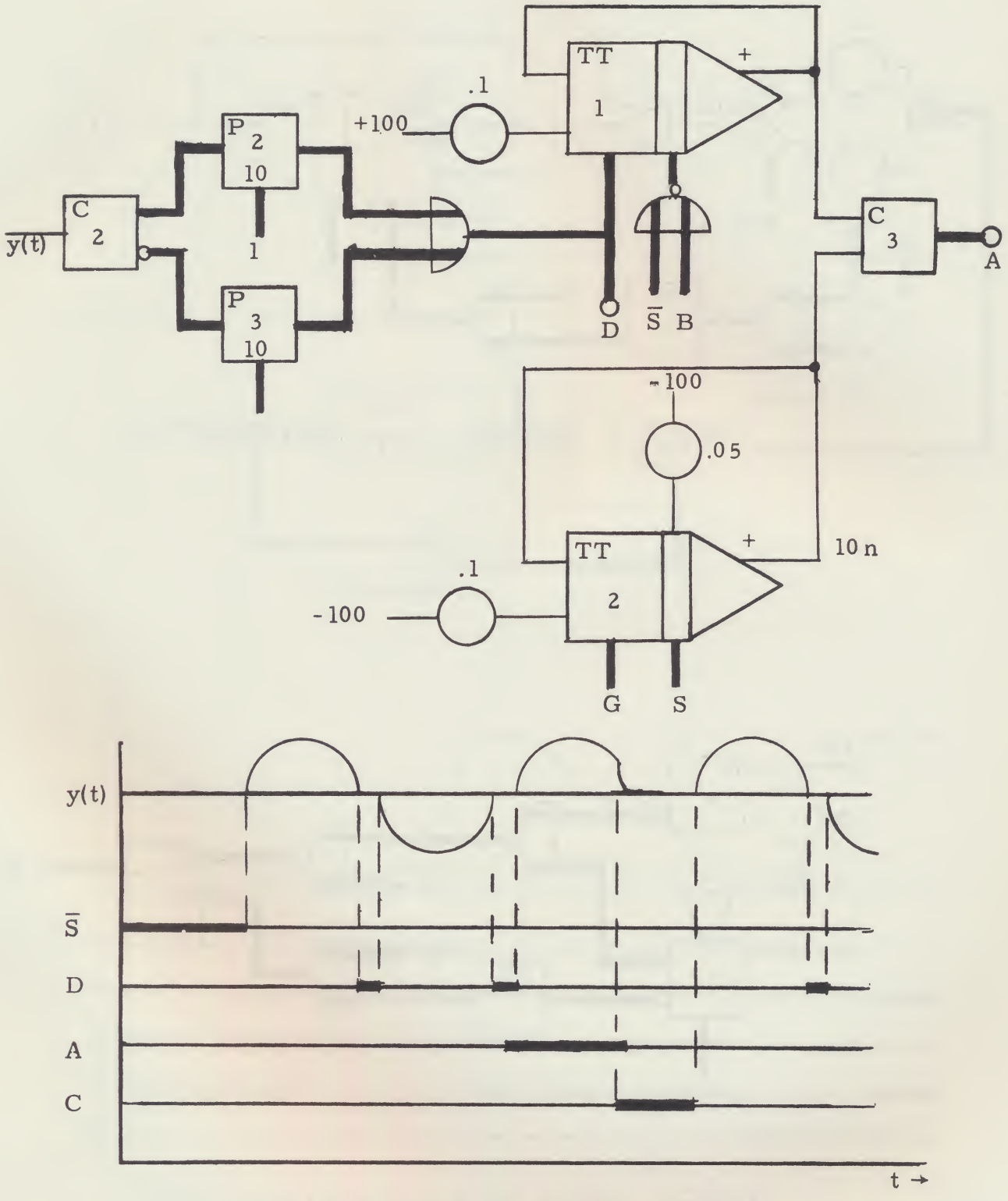

Figure 4. Zero-Crossing Counter

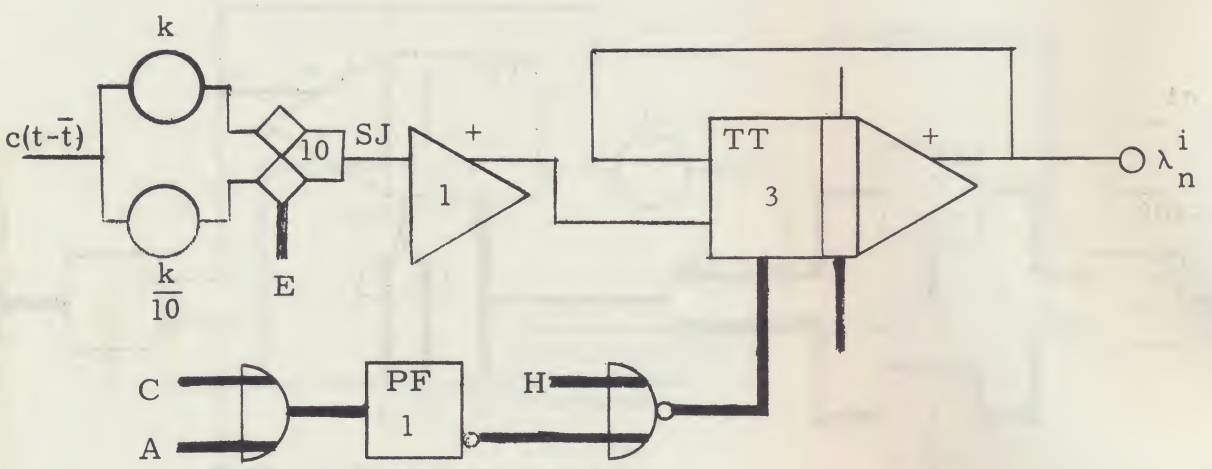

Figure 5. Parameter Correction Circuit

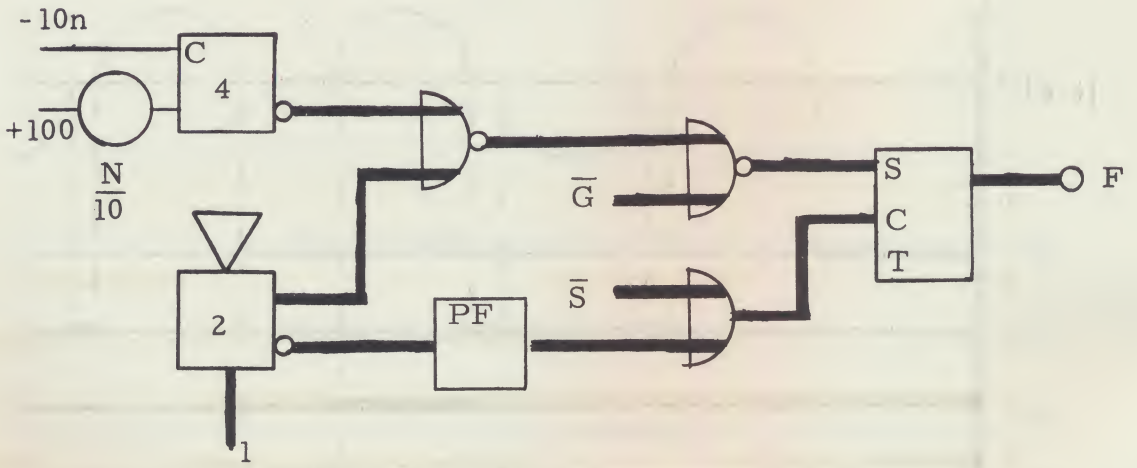

Figure 6. Manual and Programmed Stop Circuit

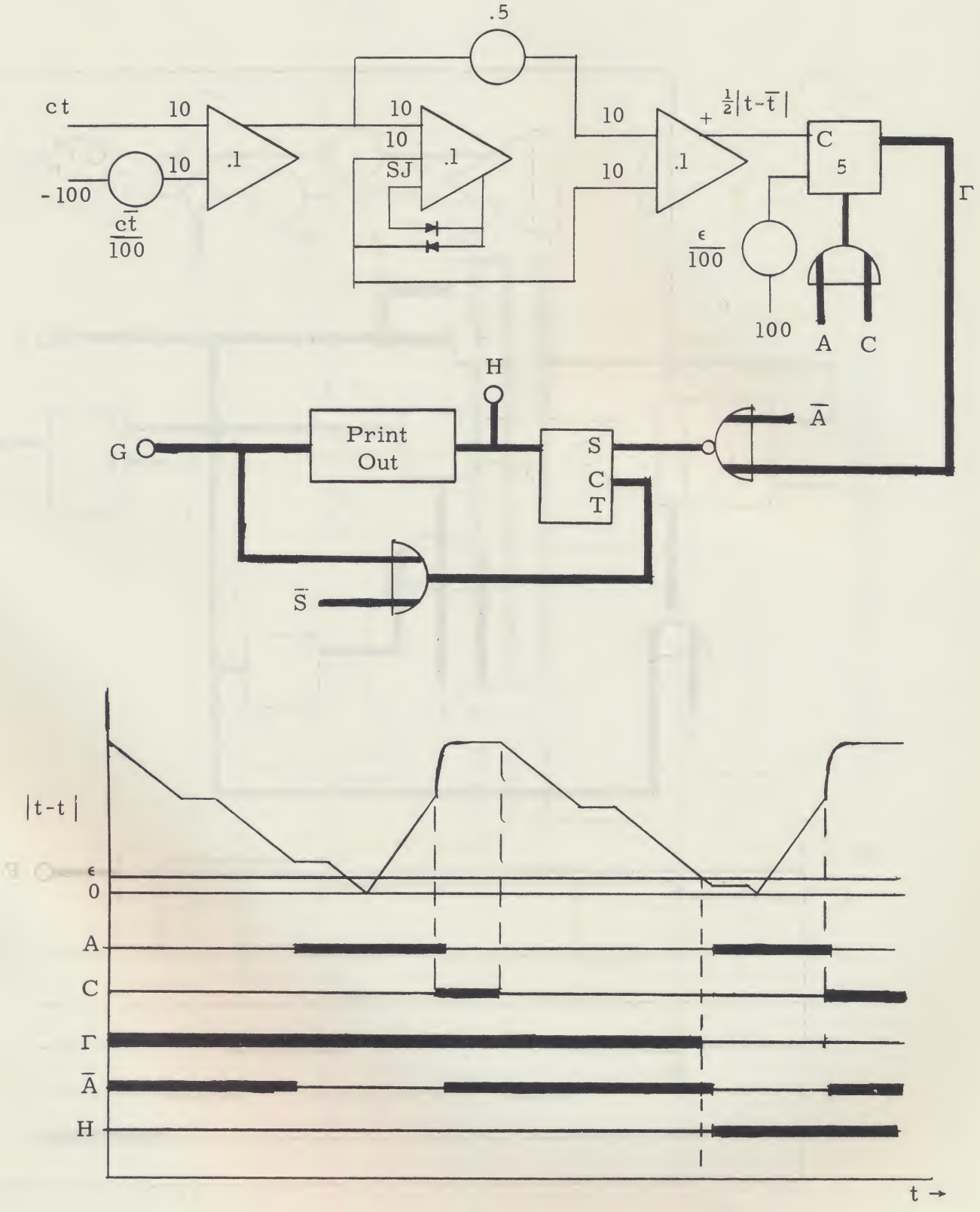

Figure 7. End Point Error Test Circuit

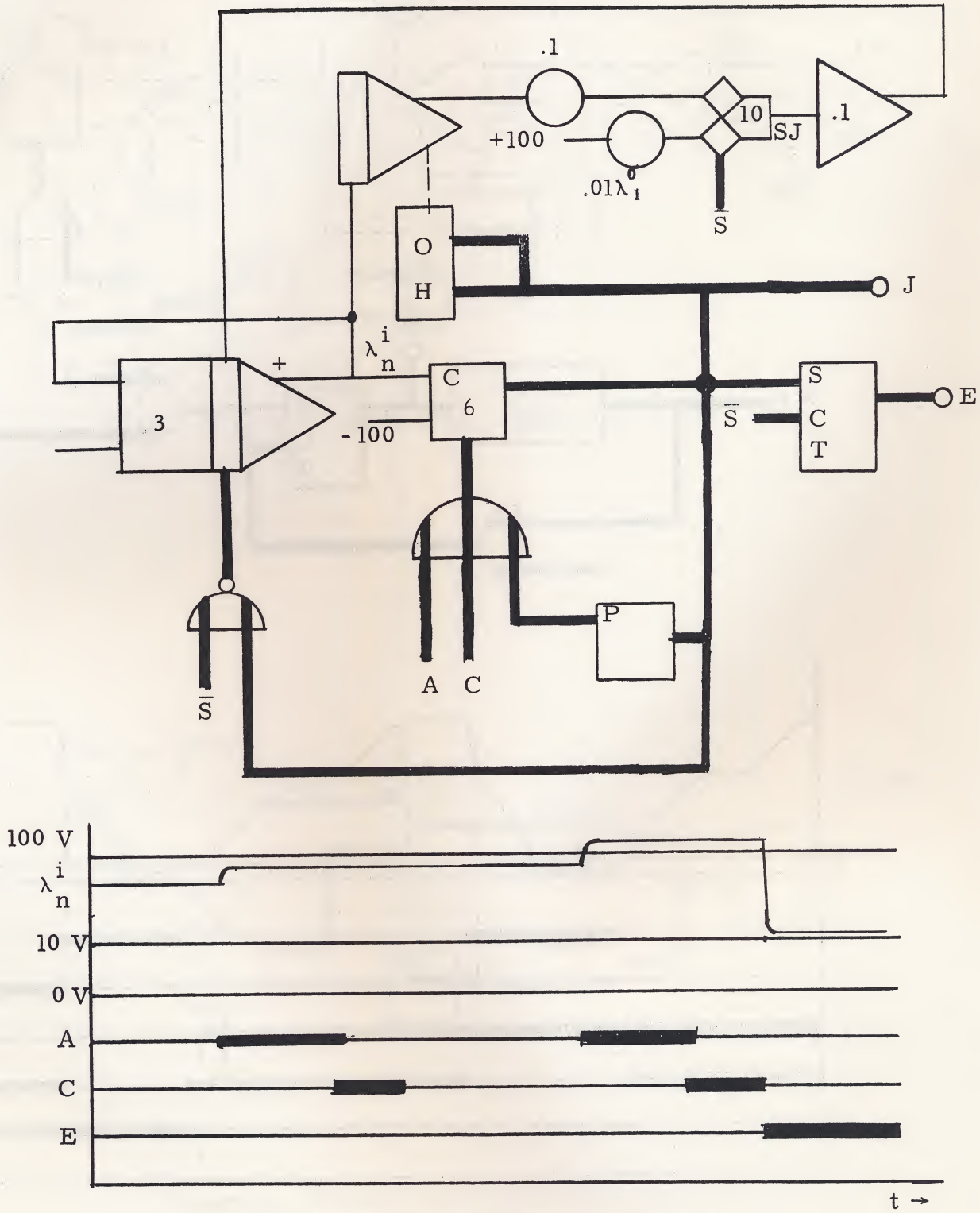

Figure 8. Automatic Scale Changing Circuit

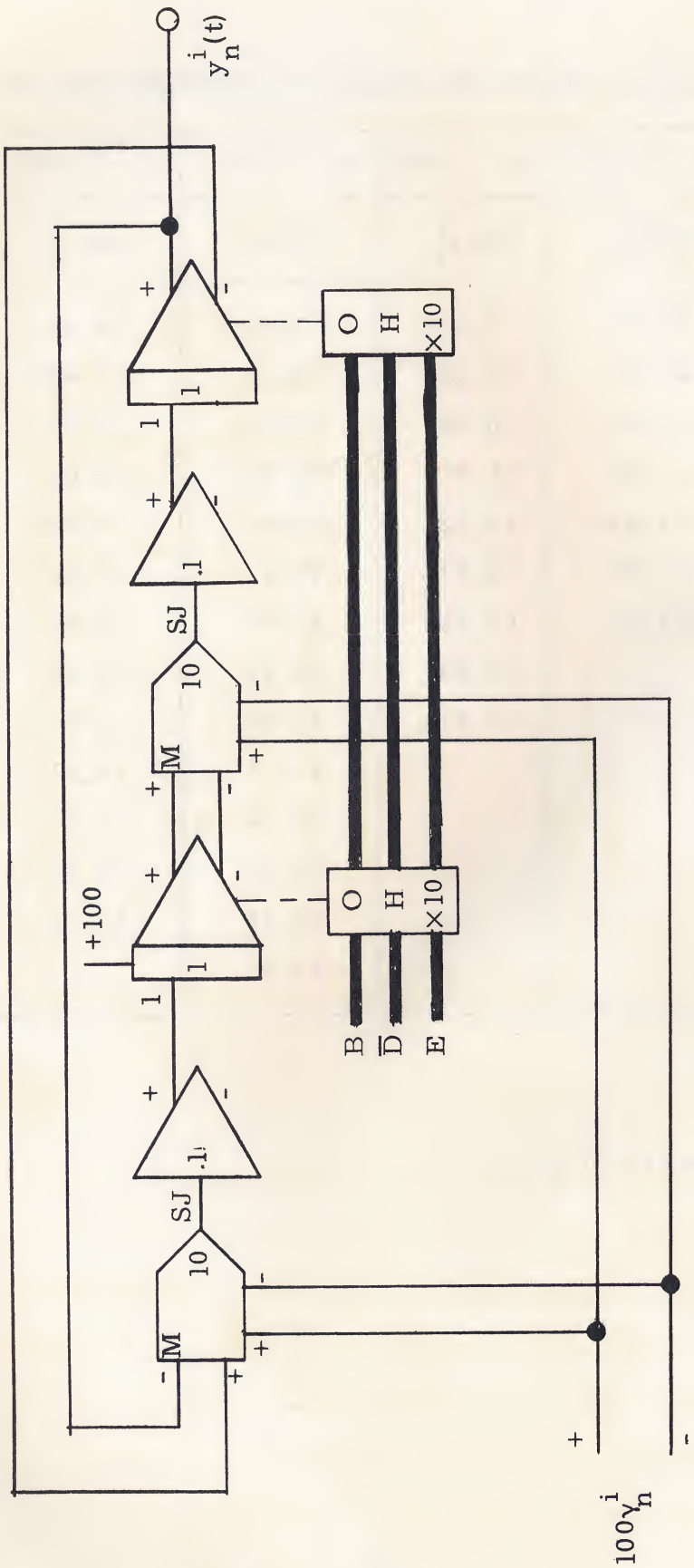

Figure 9. Uniform Slab Simulation

Table 1. Numerical Results for Homogeneous Slab

$\gamma_1 = .3142$	$\gamma_2 = .6283$	$\gamma_3 = .9425$	$\gamma_4 = 1.2566$
$100\ \gamma_1^i$	$100\ \gamma_2^i$	$100\ \gamma_3^i$	$100\ \gamma_4^i$
10.00	31.41	62.81	94.24
29.97	51.38	82.79	11.42*
31.89	60.29	88.32	11.82
31.29	61.97	90.99	12.07
31.44	62.52	92.42	12.24
31.40	62.71	93.21	12.34
31.41	62.78	93.66	12.41
	62.80	93.91	12.46
	62.81	94.05	12.49
		94.13	12.52
		94.18	12.53
		94.21	12.54
		94.23	12.55
		94.24	

*Rescaled to $10\ \gamma_4^i$.